Statistical Modeling and Latency Optimization for Entanglement Routing in Quantum Networks

Jiyao Liu and Yu Wang
Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA
{jiyao.liu,wangyu}@temple.edu

***Abstract*—Latency optimization in quantum networks is a critical challenge due to the inherent fragility of quantum operations and the exponential decay of qubits in quantum memory. Unlike classical networks, quantum networks frequently encounter failures during entanglement swapping, a fundamental operation required for establishing long-distance quantum connections. These failures introduce significant latency, complicating the end-to-end (E2E) entanglement distribution analysis. In this paper, we address the latency minimization problem in quantum networks at both the path and network levels. We model the stochastic behavior of entanglement swappings and provide a more accurate statistical representation of E2E latency. By proving the optimal substructure of E2E latency, we propose a Dynamic Programming (DP) algorithm to determine the optimal swapping order for path-level latency minimization. Then leveraging such path-level solution, we further formulate and solve a path-selection based optimization for network-wide latency routing problem as well. To further reduce latency, we also explore the benefits of redundant entanglement resources and propose a greedy algorithm to allocate these resources effectively. Through extensive simulations, we demonstrate the accuracy of our statistical modeling and analysis, the benefits of additional resources in reducing latency, and the superior performance of our proposed solutions over existing approaches at both the path and network levels.**

***Index Terms*—Entanglement Swapping, Entanglement Routing, Latency Optimization, Quantum Network**

I. Introduction

The performance metrics for quantum network services share commonalities with those of classical networks, however, latency is especially critical in quantum networks [1]–[4]. Latency not only affects the quality and reliability of entanglement connections but is also crucial for applications dependent on timely quantum communication. This is because quantum bits (qubits) stored in quantum memory suffer from decoherence, degrading exponentially over time. Besides, the quality of the end-to-end (E2E) entanglement itself is also affected by the in-memory time of the crude link-level entanglements. As a result, minimizing latency in quantum networks is significantly more important than in classical counterparts. In this paper, we focus on E2E latency minimization, addressing both path-level and network-level optimization challenges in quantum networks (QNs).

A major contributor to latency in quantum networks is operation failures. Unlike classical networks, where connection failures are rare, quantum networks experience failures frequently. The most basic operation "*entanglement swapping*" can fail with a relevantly high probability. The concrete success rate of this fundamental operation depends largely on the implementation. For example, classic linear optics-based Bell State Measurement (BSM, the key component of swapping gates) can succeed at no more than 50% [5] because it can only discern 2 out of 4 Bell states. Further improvements help it increase the probability to up to 62.5% [6] and 75% [7]. In-memory or deterministic BSM [8] typically does not have such an upper bound so it can theoretically achieve 100% success rate (ignoring noise), but it is less efficient compared to linear optics, making it less favorable for large-scale quantum networks. In this work, we consider the success probability of entanglement swapping can range between $[0.5, 1]$. Once a swapping operation fails, all previous efforts are wasted, requiring the process to restart from the elementary links. These frequent failures and retries make latency minimization in quantum networks particularly challenging and fundamentally different from its classical counterpart.

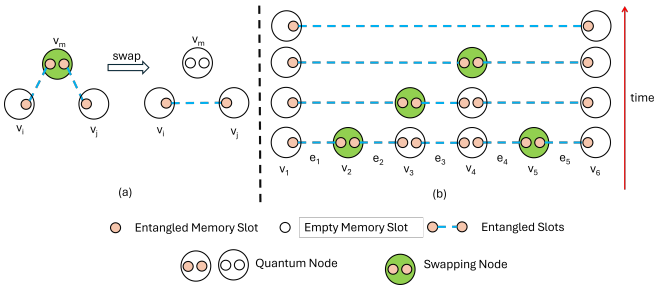


Fig. 1. **Entanglement swapping**: (a) a swapping operation over node v_m consumes two entanglements between (v_i, v_m) and (v_m, v_j) to establish a new entanglement between (v_i, v_j); (b) a sequence of swappings to establish an E2E entanglement on a 5-hop quantum path.

Fig. 1(a) shows a general swapping operation that takes two adjacent entanglements between node pairs (v_i, v_m) and (v_m, v_j) as input, and outputs a longer entanglement between (v_i, v_j). In this paper, we call that two entanglements are adjacent if they share one (and only one) common node, e.g., v_m for adjacent pairs (v_i, v_m) and (v_m, v_j). After the swapping operation, slots on nodes v_i and v_j are entangled and the two on v_m can be recycled. Such swapping operations can be performed multiple times over a quantum path (i.e., a repeater chain) to establish one E2E entanglement, as illustrated in Fig. 1(b). In this example, we aim to establish an E2E entanglement along a 5-hop quantum path, i.e., entanglement between v_1 and v_6. Two swappings are performed at v_2 and v_5 first. After these swappings, memory slots on these two nodes are released, thus we do not show them on the path anymore. Then via swappings at v_3 and v_4 in the sequential order, an E2E entanglement between (v_1, v_6) is established.

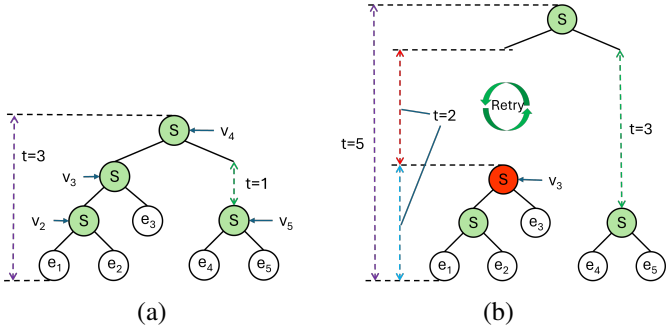


Fig. 2. **Swapping trees**: (a) an example of swapping tree corresponding to Fig. 1(b); without failures, it takes 3 steps to establish the E2E entanglement. (b) an example of swapping tree when the swapping on node v_3 fails. Note that when the swapping fails, we have to re-do swappings to build the left subpath again, thus the overall E2E latency at the root increases from 3 to 5.

Because any of those swappings may fail, it is non-trivial to characterize and optimize the E2E latency, even on a single path. Once a failure happens, we have to retry, i.e., prepare another two entanglements on the two subpaths, and try swapping again. This makes it difficult to characterize and optimize the latency. Fig. 2(a) shows a tree modeling of such swappings over a path: each link is a leaf and each intermediate node (performing one swapping) is an internal node in the tree. Then, the latency to obtain one E2E entanglement is the height of the tree. However, when any swapping fails, this modeling is inadequate. Fig. 2(b) shows an example when the swapping on v_3 fails. In that case, we have to try the left subpath again, which causes additional latency that equals to the latency of the left subpath. Actually, the retry can fail again, which makes the analysis even more complex. More importantly, the order of swappings, i.e., the shape of the swapping tree, also plays a critical role in the E2E latency. Existing literature on entanglement distribution or routing [9]–[25] still does not fully investigate and solve this problem. Note that, though we only show how to establish one single E2E entanglement, the same strategy can be used for all entanglements along the same path, no matter maintaining entanglements usable anytime (so requests can be served immediately) or establishing entanglements on-demand. As long as the order is fixed, the E2E latency always follows the same distribution.

To this end, we first characterize the statistics (such as Probability Mass Function (PMF) of latency) of a general swapping operation, and then extend this characterization to describe all swapping operations along a given path. This enables a statistical model for any node within an arbitrary swapping sequence (i.e., a swapping tree). We formally define the optimal swapping order for a given quantum path and introduce two key latency optimization problems: the path-level Optimal Latency Swapping (OLS) problem and the network-wide Optimal Latency Routing (OLR) problem. For the OLS problem, we define and prove the optimal substructure of E2E latency, which enables us to employ a Dynamic Programming (DP) algorithm to find the optimal solution. For the OLR problem, we formulate a new path-selection based variation (OLR-P), enabling us to combine our path-level DP method with a classical Integer Linear Programming (ILP) solver to derive an efficient solution. Additionally, we demonstrate that the E2E latency can be further reduced by allocating additional resources. To this end, we devise a greedy algorithm that optimizes latency at the price of additional cost. Finally, we evaluate our DP-based solution through extensive simulations, comparing it with existing path-based methods. Results confirm that our approach achieves significantly lower latency at both the path and network levels.

In summary, our contributions are threefold:

1) We introduce latency optimization problems in entanglement routing based on a stochastic model of swapping operations. We prove the optimal substructure of E2E latency along a quantum path and develop a DP algorithm to find the optimal swapping order to achieve optimal latency. This approach also enables path-selection-based latency optimization OLR-P across the network.
2) We demonstrate that allocating redundant resources can further reduce latency and propose a greedy algorithm to determine the optimal allocation of these resources.
3) Through extensive simulations, we show that our DP algorithm achieves significantly lower latency compared to existing path-based swapping methods. Furthermore, redundant entanglement provisioning can further reduce latency at both the path and network levels.

II. Related Works

In this section, we mainly review the recent results on the latency analysis of quantum repeater chains [11]–[14] and networks [15]–[17]. Note that there are other works on entanglement distribution [26]–[29].

Latency Analysis for Repeater Chains. The exact latency over a general multi-hop repeater chain remains an open problem, even when each edge comprises only a single quantum channel [11]. Some studies [11]–[13] aim to approximate the probability distribution of latency. For example, Ghaderibaneh *et al.* [13] employ a dynamic programming (DP) method to identify swapping trees with optimal latency by approximating the geometric latency distribution with an exponential distribution. This approximation is accurate only when the swapping success probability is close to zero [11]. However, many studies suggest that swapping success probabilities can reach or exceed 50% [5]–[8], as discussed in Section-I. Furthermore, most existing works [11]–[13] assume each edge has only one single channel, whereas QNs commonly utilize multiple channels per edge to enhance performance [9], [10], [30]. This discrepancy limits the applicability of these analyses to practical QNs. [14] accommodates multiple channels per edge but only considers the sequential swapping order, which is empirically shown to be suboptimal [15] and theoretically proven to be the worst [25] for various metrics. Our simulation results in Section-VII further confirm that sequential swapping yields the highest latency compared to other swapping orders.

Latency Optimization in Quantum Networks. Network-wide latency optimization [15], [16] typically assumes that the available entanglements or generation rates at the edge level

are given, focusing the optimization efforts on network-wide operations. [15] simplifies latency by treating the hop count of a path as a proxy for latency, selecting shorter paths to achieve lower latency. While this approach is reasonable, it falls short of accurately quantifying actual latency. In contrast, our work rigorously models the stochastic behavior of swapping operations along a path and provides a precise PMF for E2E latency. [16] formulates the problem as a multi-commodity flow scheduling problem over multiple time intervals, aiming to prioritize requests with deadlines to ensure that they are served on time. This approach emphasizes resource competition among user demands. By contrast, our work focuses on reducing latency for individual requests and minimizing total latency across all requests given the available resources, rather than solely addressing resource competition. [17] proposes opportunistic routing with analysis to reduce network-wide latency but assumes one single channel per edge, leading to limitations similar to those seen in works on repeater chains [11]–[13]. In contrast, our approach aligns with the assumptions in [15], maintaining flexibility for edge hardware configurations. Finally, none of the above works [15]–[17] consider utilizing redundant resources to reduce latency — a capability that distinguishes our approach.

Notably, [31] also leverages redundant resources to improve fault tolerance in entanglement distribution within QNs. However, their approach focuses on utilizing additional or idle links to increase the probability of successful entanglement creation, aiming to maximize throughput. This objective differs fundamentally from our focus on minimizing E2E latency.

III. Network Model and Problem Definition

In this section, we first introduce our latency and cost modeling on swapping operations over paths, then formally define the optimal latency swapping problem over a single path and the optimal latency routing problem in the QN.

A. Swapping and Tree Modeling

Swappings are used to connect two adjacent entanglements to form a longer one, such as in Fig. 1(b), the swapping on the node v connects two entanglements between nodes (i, v) and (v, j) to form one entangled pair between (i, j). Over a multi-hop quantum path, this operation can be repeated multiple times to obtain an E2E pair as shown in Fig. 1. Swappings can be either probabilistic (which may fail) or deterministic, depending on the implementations. Once failed, the input entanglements get lost without any outcome. Deterministic swappings do not have such upper bounds, but they are typically less efficient (in terms of processing ability) compared to linear optic ones, making them less suitable for large-scale quantum networks. We allow the success probability in $[0.5, 1]$ to cover all these techniques. It is obvious that the failures of swappings are one of the dominating reasons for E2E latency and cost in entanglement routing.

Tree-based modelings are proven effective for describing many characters of swapping operations over quantum paths [12], [13]: the swappings take two adjacent entanglements as input, and output a 'merged' entanglement, so it is easy to see that we can model this process by assigning to two input entanglements as children of the output entanglement. We can use a binary tree to represent the relation of all swappings over a path as shown in Fig. 2. Such a swapping tree describes an arbitrary swapping order over a path. As the order of swappings significantly affects the latency, to facilitate latency optimization, we now formally define a swapping sequence (serialized swapping order) of a path π.

***Definition** 1:* A *swapping sequence* λ_π is the post-order traversal of a swapping tree over path π.

Here, we use the post-order because, in nature, we have to wait until the two child swappings are completed before we can execute the current swapping. By using post-order, we make sure that all input swappings are before the output swappings in the sequence. For example, the sequence of the swappings shown in Fig. 2 is (v_2, v_3, v_5, v_4).

B. Latency and Cost Characterization of Swappings

Now we can model the latency and cost to establish one E2E entanglement over a quantum path. We first model the latency/cost incurred by a general swapping operation, then we can easily stack it up to obtain the E2E latency/cost. For a swapping on node v to connect left and right path fractions (i, v) and (v, j) to form an entanglement on (i, j), we can model the time needed to obtain an output by the following random variables

$$L_v = (W_v + t) \cdot S_v, \quad W_v = \max(L_l, L_r).$$

Here, L_v is the time needed to perform a successful swapping at node v to obtain an entanglement pair on (i, j), while W_v is the waiting time at node v before a swapping operation.

L_l and L_r are random variables indicating the latency to establish the left and right sub-paths (as input), respectively. Note that only when both sides are ready, we can do a swapping at node v. Recall that we focus on the network scale scheduling so we assume the physical/link layer protocols are always trying to establish and maintain as many entanglements as possible, so the network scheduler can immediately access link-level entanglements (if available). That is, in the base cases where the subpaths are single-hop quantum edges (leaves in the swapping tree), we set $L_e = 0$. In a general case, L_l and L_r are obtained from previous swappings, which are solved in previous steps.

To obtain L_v, besides waiting time W_v we also consider the operation time t required by a swapping operation and the number of swapping attempts S_v to conduct one swapping successfully on node v. S_v is a geometric distribution parameterized by the success probability q_v of swapping on node v, i.e., $S_v \sim G(q_v)$.

Similarly, we can derive the cost (i.e., the number of elementary entanglements used) of a successful swapping at node v from the cost of subpaths. Obviously, each swapping attempt consumes one entanglement on each subpath, and we should do swapping until it succeeds once. Suppose C_l and C_r are the random variables denoting the cost of left and right subpaths, the total cost C_v at v can be modeled as

$$C_v = (C_l + C_r) \cdot S_v.$$

Cost can also be used as a type of optimization criterion when searching for good swapping orders, e.g., for those orders using the least entanglements on the path. We will use cost as the criterion in a variation of our later proposed algorithm.

C. Optimal Swapping Order over Quantum Path

Since different swapping sequences may lead to different latency to establish one E2E entanglement over a single path and the achieved latency is also probabilistic even for the fixed swapping order, we are interested in finding the optimal swapping order to minimize the achieved probabilistic latency. To define the optimal swapping order, we first introduce two concepts to compare random variables.

***Definition** 2:* Random variable X_1 *stochastically dominates* X_2 if $F_{X_1}(x) \leq F_{X_2}(x)$ for any x. We denote this relation by $X_1 \geq_{\text{st}} X_2$ and $X_2 \leq_{\text{st}} X_1$. Here, $F_X(x)$ is the cumulative distribution function (CDF) of random variable X.

***Definition** 3:* Random variable X_1 is *smaller than or equal to* X_2 in expectation if $E[X_1] \leq E[X_2]$. We denote this relation by $X_1 \leq_{\text{exp}} X_2$ and $X_2 \geq_{\text{exp}} X_1$.

With $\leq_{\text{st}}$ and $\leq_{\text{exp}}$, we can also define $\min_{\text{st}}$, $\max_{\text{st}}$, $\min_{\text{exp}}$ and $\max_{\text{exp}}$ operators on a set of random variables accordingly. For example, $\min_{\text{st}}$ picks the one that is dominated by all others in a set of random variables. These operations will be useful in searching for the optimal swapping solution. An important and straightforward implication is that if $X_1 \leq_{\text{st}} X_2$, then $X_1 \leq_{\text{exp}} X_2$, but it is not necessarily true reversely.

Intuitively, we prefer the optimal swapping solution that gives us the lowest latency in expectation, but we can prove a better bound, i.e., the optimal order not only has the lowest average latency, but also is stochastically dominated by all other solutions. This is more desirable because stochastic dominance also gives us the quality of service guarantee, i.e., given any latency requirement, the optimal solution always has a higher probability of providing an E2E entanglement within the time. This is not true by defining the optimal solution as the one with the lowest expected latency. We will later see that it is not easy to find the swapping order with the lowest expected latency directly without using stochastic dominance.

Now, we define the *optimal swapping order* for a given quantum path π which consists of vertices $v_1 \leftrightarrow v_{|\pi|}$ with $|\pi|-1$ edges $\{e_1, e_2, ..., e_{|\pi|-1}\}$, where the edge e_i connects nodes v_i and v_{i+1}. Suppose Λ_π is the set of all possible swapping orders for path π and $L^{\text{E2E}}_{\lambda_\pi}$ (as a random variable) is the E2E latency on path π achieved by the swapping order λ_π. Then the optimal swapping order λ^*_π is defined as follows.

***Definition** 4:* **Optimal Swapping Order.** A swapping order λ^*_π is *optimal* if and only if $L^{\text{E2E}}_{\lambda^*_\pi} \leq_{\text{st}} L^{\text{E2E}}_{\lambda_\pi}$ for $\forall \lambda_\pi \in \Lambda_\pi$.

Then, it is straightforward that our problem on one single path would be finding the optimal solution with lowest latency:

***Problem** 1:* **Optimal Latency Swapping (OLS) Problem.** Given a quantum path π, find the optimal swapping order λ^*_π such that $L^{\text{E2E}}_{\lambda^*_\pi} \leq_{\text{st}} L^{\text{E2E}}_{\lambda_\pi}$ for $\forall \lambda_\pi \in \Lambda_\pi$.

D. Optimal Latency Routing in Quantum Network

With the definition of optimal swapping order over a path, we can then consider the overall network-wide latency minimization of entanglement routing in a quantum network. Consider a quantum network $G(V, E)$, where V and E are the set of nodes (quantum repeaters with quantum memory) and edges (quantum links). We assume that a node v has a limited m_v of memory slots and a link e has up to c_e generated entanglements during the duration of each round of routing. We consider a set of user pairs U demanding entanglements, each user pair $u = (v_i, v_j)$ requires d_u entanglements between v_i and v_j. We denote the demand set as $D = \{d_u | \forall u \in U\}$. We use P to denote all simple paths between all user pairs in the network. The goal of the routing problem is to find paths from P and their swapping orders for all user demands so that (i) the user demands are satisfied, (ii) those selected paths do not use more than available resources, and (iii) the total expected latency of all paths is minimized.

***Problem** 2:* **Optimal Latency Routing (OLR) Problem**. Given network $G(V, E)$ and user demands D, find a subset of paths $P^* \in P$ and their swapping schemes $\{\lambda_\pi | \forall \pi \in P^*\}$ so that the total latency in expectation is minimized.

Obviously, we may have $O(|V|!)$ possible paths for each user pair in G. It is expensive to exhaust all $O(|U| \cdot |V|!)$ paths for large networks to find the optimal solution of the OLR problem. Therefore, in this paper, we consider a path-based optimization, where we limit the candidate paths between each user pair u (denoted by P_u) to a constant size. Obviously, this simplification sacrifices the optimality of the original OLR problem for efficiency. The following is a formal formulation for this new **Path-based OLR (OLR-P) Problem**.

$$\textbf{OLR-P}: \min_{x} \quad \sum_{u \in U, p \in P_u} t_{u,p} \cdot x_{u,p} \tag{1}$$

s.t. entanglement requirement

$$\sum_{p \in P_u} x_{u,p} \geq d_u, \qquad \forall u \in U \tag{2}$$

resource constraints

$$\sum_{u \in U, p \in P_u, e \in p} \alpha_{u,p,e} x_{u,p} \leq c_e, \qquad \forall e \in E \tag{3}$$

$$\sum_{\substack{u \in U, p \in P_u \\ e \in p \cap e \in adj(v)}} x_{u,p} \leq m_v, \qquad \forall v \in V \tag{4}$$

decision variables

$$x_{u,p} \in \mathbb{Z}^+_0. \tag{5}$$

The only decision variable $x_{u,p}$ denotes the number of entanglements generated for user pair u via a candidate path $p \in P_u$. Objective (1) simply aims to minimize the summation of latency of all selected paths. Here, $t_{u,p}$ is the expected latency for path p between user pair u, which is an input variable obtained by a path-level swapping scheme for the OLS problem (as will be discussed in Section IV). Constraint (2) ensures that the generated entanglements between each pair are no less than the demand. Constraints (3) and (4) limit the used entanglements and memory slots to not exceed the available resources at each edge and node. $\alpha_{u,p,e}$ is the expected number of entanglements used on edge e by the path

p between user pair u, obtained by the path-level swapping scheme (and its calculation will be presented in Section VI).

Generally, OLR-P is a path selection problem: among all candidate paths between each user pair, we need to select some paths and use them to suffice the user demands. Both $t_{u,p}$ and $\alpha_{u,p,e}$ are inputs for the problem, while they are the results from the path-level swapping solutions for the OLS problem. Obviously, they have a critical impact on the solution of this OLR-P problem. OLR-P itself is an integer linear problem, which can be solved by a classical ILP solver. Therefore, our remaining focus is solving the path-level OLS problem.

IV. Optimal Latency Swapping: Latency Analysis and DP-based Swapping Scheme

In this section, we show that the optimal swapping order of the OLS problem can be found via a dynamic programming (DP) algorithm. Fig. 3 illustrates the basic idea of our DP algorithm using an example. Given a path fraction v_1 to v_6, it should exhaust and compare all possible swapping nodes (v_2, v_3, v_4, and v_5). For each node, it calculates the statistics of output entanglement using statistics of input entanglements, which are calculated in previous rounds. After having the statistics (e.g., PMFs) of all output entanglements, it picks the best one as the solution for this path fraction, and proceeds to the next level (solving longer paths). Fig. 3 shows two possible splits (performing swapping at nodes v_3 and v_4) and their swapping trees of output entanglements.

Such a DP-based solution is non-trivial because (i) it is not straightforward to see that the OLS problem has an optimal substructure, without which the DP method may not be able to find the optimal solution; (ii) even if it has an optimal substructure, we need the recursion formula that correctly calculates the statistics of random variables, which is rare in the general DP design.

A. Proof of Optimal Substructure

To elaborate on the details of our DP design, we first introduce a seemingly viable but actually incorrect DP formula, where the optimal solution is defined by expected latency. At node v, with the definition of L_v, it is easy to see

$$E[L_v] = (E[W_v] + t) \cdot E[S_v] = \frac{1}{q_v}(E[W_v] + t).$$

This means that $E[L_v]$ is linear to $E[W_v]$, so obviously optimal W_v leads to optimal L_v. If breaking W_v into $E[L_l]$ and $E[L_r]$, we may try to obtain the recursion formula as

$$E[L_v] = \min_{i<v<j} \frac{1}{q_v}[\max(E[L_l], E[L_r]) + t].$$

However, this formula is actually wrong because of the non-linearity of max of the expectation of two random variables. That is, $E[W_v] = E[\max(L_l, L_r)] \neq \max(E[L_l], E[L_r])$, which makes the above formula incorrect. Unfortunately, it is hard to express $E[L_v]$ solely by $E[L_r]$ and $E[L_r]$, especially when they represent the results of multiple previous swappings (so $E[L_v]$ is a non-standard distribution). This is also the reason why it is challenging to directly prove the optimal substructure when defining the optimal solution by expectation.

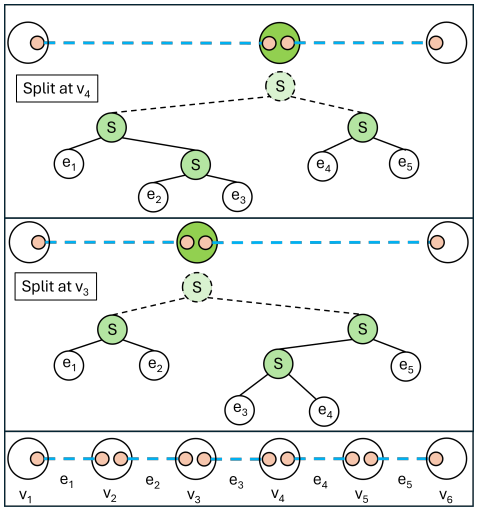


Fig. 3. **Idea of DP algorithm**: path fraction from v_1 to v_6, and two possible splits (one at v_3 and the other at v_4)with their corresponding swapping trees.

Now, we show that L_v has optimal substructure under Definition 4 by proving Theorem 1. Suppose λ_l^* and λ_r^* are the optimal swapping orders of the subpaths l and r of path p (and they share the same node v), Λ_π^v is the set of all swapping orders that performs the last swapping on node v (i.e., $\Lambda_\pi^v = \{(\Lambda_l, \Lambda_r, v) | \forall \Lambda_l, \Lambda_r\}$), and $\lambda_\pi^* = (\lambda_l^*, \lambda_r^*, v)$, then we can show L_{λ_π} has the optimal substructure property by the following theorem.

***Theorem** 1:* If $L_{\lambda_l^*} \leq_{\text{st}} L_{\lambda_l}$ for $\forall \lambda_l \in \Lambda_l$ and $L_{\lambda_r^*} \leq_{\text{st}} L_{\lambda_r}$ for $\forall \lambda_r \in \Lambda_r$, then $L_{\lambda_\pi^*} \leq_{\text{st}} L_{\lambda_\pi}$ for $\forall \lambda_\pi \in \Lambda_\pi^v$.

Proof: We first prove that W_v has optimal substructure regarding L_l and L_r at node v, then prove that L_v has optimal substructure regarding W_v. Hereafter, when the swapping node v is clear from context, we drop it from the notation of L_v, W_v, and S_v for simplicity. Let $W^* = \max(L_{\lambda_l^*}, L_{\lambda_r^*})$ and recall that $F_X(x)$ is the CDF of X.

By the order statistics, we have

$$\begin{aligned} F_W(w) &= \Pr\{W \leq w\} = \Pr\{\max(L_{\lambda_l}, L_{\lambda_r}) \leq w\} \\ &= \Pr\{L_{\lambda_l} \leq w, L_{\lambda_r} \leq w\} \\ &= \Pr\{L_{\lambda_l} \leq w\} \Pr\{L_{\lambda_r} \leq w\} \\ &= F_{L_{\lambda_l}}(w) \cdot F_{L_{\lambda_r}}(w). \end{aligned} \tag{6}$$

Then, given $F_{L_{\lambda_l^*}}(k) \geq F_{L_{\lambda_l}}(k)$ and $F_{L_{\lambda_r^*}}(k) \geq F_{L_{\lambda_r^*}}(k)$,

$$\begin{aligned} F_{W^*}(w) &= F_{L_{\lambda_l^*}}(w) F_{L_{\lambda_r^*}}(w) \\ &\geq F_{L_{\lambda_l}}(w) F_{L_{\lambda_r}}(w) \\ &= F_W(w). \end{aligned} \tag{7}$$

That is, $W^* \leq_{\text{st}} W$, i.e., W has suboptimal structure.

Now we prove that L has optimal substructure regarding W, i.e., $L_{\lambda_\pi^*} \leq_{\text{st}} L_{\lambda_\pi}$ if $W^* \leq_{\text{st}} W$. For any λ_π, we have

$$\begin{aligned} F_{L_{\lambda_\pi}}(k) &= \Pr\{L \leq k\} = \Pr\{(W + t) * S \leq k\} \\ &= \sum_{s \in Z^+} \Pr\{(W + t) \cdot S \leq k | S = s\} \Pr\{S = s\} \end{aligned}$$

$$= \sum_{s \in Z^+} \Pr\{W \leq \frac{k}{s} - t\} \Pr\{S = s\}$$
$$= \sum_{s \in Z^+} F_W(\frac{k}{s} - t) f_S(s). \quad (8)$$

Because $F_{W^*}(w) \geq F_W(w)$ is true for any w as we have proven $W^* \leq_{\text{st}} W$, it is easy to see that $F_{L_{\lambda_\pi^*}}(k) \geq F_{L_{\lambda_\pi}}(k)$, i.e., $L_{\lambda_\pi^*} \leq_{\text{st}} L_{\lambda_\pi}$. ∎

This theorem means that if the solutions to the two subpaths are optimal, then the solution of the whole path is also optimal among all solutions whose last swapping happens on v (i.e., Λ_π^v). By iterating all possible v with a DP method, we can find the optimal solution of OLS among $\Lambda_\pi = \cup_v \Lambda_\pi^v$.

B. DP Calculation

We have proven that a DP algorithm can find the optimal solution, but still need to find how to calculate L's distribution given L_l and L_r. It is not straightforward to see an analytical relation on their statistics, such as PMF $f_X(x)$ or CDF $F_X(x)$, because L is likely not a standard well-studied distribution, especially when the inputs are from previous swappings. To complete the calculation, we introduce a precision factor of K to cut off the tails of PMF/CDF. If not, the complete PMF and CDF are defined on the infinite Z^+ that makes numerical calculation impossible. Actually, the precision factor does not affect the accuracy of $F_L(k)$ for $\forall k \leq K$. This is because we can find F_L by

$$\begin{aligned} F_L(k) &= \Pr\{(W+t)S \leq k\} \\ &=_a \sum_{w=1}^{K} \sum_{s=1}^{K} \Pr\{(W+t)S \leq k | W = w, S = s\} \\ &= \sum_{w=1}^{\lfloor k-t \rfloor} \sum_{s=1}^{\lfloor k/(w+t) \rfloor} f_W(w) f_S(s). \end{aligned} \quad (9)$$

In step $=_a$ we do not need to consider $w > K$ or $s > K$ because we known $\Pr\{(W+t)S \leq k\} = 0$ for $\forall k \geq K$ in these cases. Since S is known as input, we only need f_W, which can be derived as

$$f_W(w) = F_W(w) - F_W(w-1), \quad (10)$$

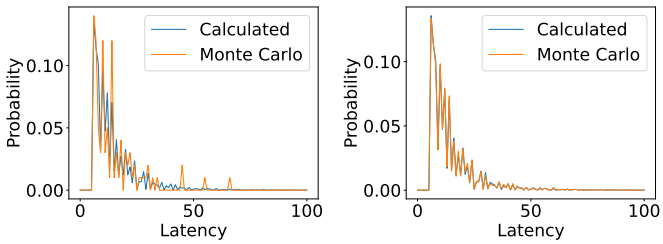
where we can define $F_W(-1) = 0$, and calculate $F_W(w)$ for $\forall w \leq K$ by Eq. (6). Till now, we obtain $F_L(k)$ for future recursive computation. Note that $F_L(k)$ for $\forall k \leq K$ only rely on $F_{L_l}(k)$ and $F_{L_r}(k)$ for $\forall k \leq K$. That is, cutting off the tails of CDFs in our DP recursion does not affect the accuracy of $F_L(k)$ (for $\forall k \leq K$).

Now, we obtain the DP formula for the OLS problem,

$$L_{\lambda_\pi^*} = \min_{\text{st}\ i<v<j} [(\max(L_{\lambda_l^*}, L_{\lambda_r^*}) + t) \cdot S_v], \quad (11)$$

where $(\max(L_{\lambda_l^*}, L_{\lambda_r^*}) + t) \cdot S_v$ for each v gives us the optimal solution in Λ_π^v, and it is calculated for every non-ending nodes v on path π to find the optimal solution in $\Lambda_\pi = \cup_v \Lambda_\pi^v$. Again, $W = \max(L_{\lambda_l}, L_{\lambda_r})$ can be calculated by Eq. (6) for $\forall \lambda_l / \lambda_r$.

We need $\mathcal{O}(K)$ for computing F_W and f_W, $\mathcal{O}(K^2)$ to iterate over all $f_W(w)$ and $f_S(s)$ to find $F_L(k)$. That is, we need $\mathcal{O}(K^2)$ to examine one vertex as the swapping node so we use $\mathcal{O}(|\pi|K^2)$ to examine all intermediate vertices for one path fragment. We have $\mathcal{O}(|\pi|^2)$ path fragments, so the overall time complexity of DP for path π is $\mathcal{O}(|\pi|^3 K^2) = \mathcal{O}(|V|^3 K^2)$.



(a) Rounds of MC = 100 (b) Rounds of MC = 10,000

Fig. 4. PMF of latency: calculated when the precision factor $K = 100$ vs Monte-Carlo simulations with different rounds.

C. Model Validation

We now validate the proposed calculation method based on a statistical model by comparing its results to the real simulated distribution obtained via a Monte Carlo (MC) method. We pick a random swapping tree for a 15-hop quantum path to illustrate our proposed method work for an arbitrary swapping tree. Fig. 4 shows the PMF of the latency calculated by our method compared with the latency distribution by Monte Carlo simulations with different rounds.

We can have the following two observations. First, the difference between our result and MC's diminishes when increasing the simulation rounds of the MC method. This is obvious: as the simulation round increases, the distribution given by the Monte Carlo method approaches the real distribution. Second, we can confirm that the approximate factor K does not affect the accuracy of $f_L(k)$ (also $F_L(k)$) for $\forall k \leq K$. That is, our modeling method gives us exactly the real distribution in our considered regime ($k \leq K$).

V. Latency Improvement via Redundant Entanglement Provision

Using the DP-based swapping scheme, we can find the optimal latency swapping order on an arbitrary given path. However, in the case where we still have additional resources (entanglements) in the network, we may take advantage of them to further reduce latency. In the current solution, this is impossible as all the path-level solutions are already fixed so they cannot use additional resources. In this section, we introduce and model the process of using additional entanglements to further reduce E2E latency.

A. Latency Reduction from Redundant Resources

First, we show by analysis how redundant entanglement helps to further reduce latency. Suppose we have $m \geq 1$ and $n \geq 1$ entanglements on the left subpath and right subpath, respectively. Now, the time required for the left subpath is

$$L_l = \min_{i \leq m} L_l^i. \quad (12)$$

where L_l^i is the latency of the i'th entanglement on the left subpath. Now, L_l is the latency of the left subpath when we prepare m entanglements on the left subpath at the same time (rather than only 1 in previous settings). Clearly, the left path is finished as soon as we have any one of those m entanglements succeed, so the wait time for the left subpath is the minimum of those entanglements. We can calculate the CDF of L_l by

Algorithm 1 REDUNDANT SWAPPING PROVISION

Input: Stochastic swapping tree t, additional swappings budget R.

Output: Stochastic swapping trees with redundant swappings $T = \{t_r | 0 \le r \le R\}$.

$r = 1, t_0 = t, T = \{t_0\}$, ▷ result set
while $r \le R$ **do**
  $\eta_r =$ FIND-CRITICAL-SWAP(t_{r-1}) ▷ find best swap
  ENHANCE(η_r) ▷ allocate additional resource
  $t_r =$ BACKWARD(η_r) ▷ update nodes backwards
  $T = T \cup \{t_r\}$
  $r = r + 1$
return T

$$\begin{aligned} F_{L_l}(k) &= \Pr\{L_l \le k\} = 1 - \Pr\{L_l > k\} \\ &= 1 - \Pr\{L_l^1 > k, L_l^2 > k, ...\} \\ &= 1 - \prod_i \Pr\{L_l^i > k\} = 1 - \prod_i (1 - F_{L_l^i}(k)) \\ &= 1 - (1 - F_{L_i^1}(k))^m. \end{aligned} \tag{13}$$

The last step assumes that all attempts L_l^i on the left subpath are using the same swapping order, i.e., the optimal one. That is, they are all independent and identically distributed so we have $L_l^i = L_l^1$ for $\forall i \le m$. It is similar for the right subpath. Then, the latency of the whole path is

$$\begin{aligned} F_L(k) &= F_{L_l}(k) F_{L_r}(k) \\ &= \left[1 - (1 - F_{L_l^1}(k))^m\right] \left[1 - (1 - F_{L_r^1}(k))^n\right]. \end{aligned} \tag{14}$$

Clearly, when $m = 1$ and $n = 1$, this formula degrades to (6). When m and/or n increases, $F_L(k)$ increases monotonously, which means we can always reduce the latency by preparing more entanglements on the subpaths.

B. Redundant Subpath Determination

Next, we need to know which subpath should be prepared with additional pairs. It is not easy to tell the optimal general swappings to provide redundant entanglements. We propose a greedy algorithm that finds the *critical primary swapping* that reduces L^{E2E} most each time, and keeps providing copies of those critical primary swappings until a given budget of R primary swappings is consumed. Primary swappings are those that take two edges as input. Those whose inputs are derived from previous swappings are not primary. Algorithm 1 shows the overall workflow of this proposed greedy algorithm. It takes the result from the DP algorithm as input, and outputs the solutions to the same path but with a different number of additional primary swappings. In each while loop, it first finds the most critical primary swapping by FIND-CRITICAL-SWAP. Here we use η_r to denote the found critical swapping. Then, it adds one redundant swapping to it by calling ENHANCE. This change to the latency of all its ancestor swappings should be propagated till the root using BACKWARD. This is simply calculating the new statistics of ancestors with the updated information from the selected swapping to the root. Finally, include the current tree t_r to the solution list T and enter the next loop if the condition allows. FIND-CRITICAL-SWAP is the most complicated function as it actually contains ENHANCE and BACKWARD: it performs 'virtual' ENHANCE and BACKWARD for every primary swapping to see its impact on the root latency. 'virtual' means that the updates are performed in a separated buffer and thus do not affect the tree. Once found the one that reduces the root latency the most, it is selected and 'real' ENHANCE and BACKWARD are performed to update the tree.

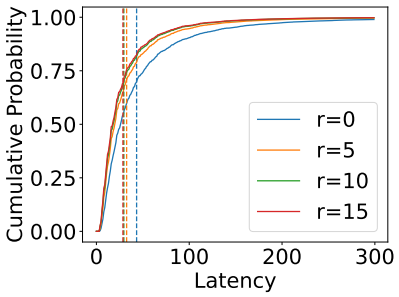


Fig. 5. CDFs of E2E latency on a quantum path with additional cost of r.

In each iteration of the while loop, we use $\mathcal{O}(|\pi|^2 K^2)$ to find the critical swapping: we have at most $\mathcal{O}(|\pi|)$ primary swappings, and we examine each primary swapping by $\mathcal{O}(|\pi|K^2)$ time. To examine one primary swapping, it uses $\mathcal{O}(|\pi|)$ time to do 'virtual' BACKWARD and uses $\mathcal{O}(K^2)$ to calculate the CDF of each ancestor. Both 'virtual' and real BACKWARD tracks back from the critical swapping to the root, so they both use $\mathcal{O}(|\pi|)$ time, which is exactly the worst height of a binary tree with $\mathcal{O}(|\pi|)$ leaves. ENHANCE can be done in $\mathcal{O}(1)$ by simply adding 1 to a counter (indicating how many copies are used for the swapping). Overall, the time complexity of Algorithm 1 is $\mathcal{O}(R|\pi|^2 K^2)$.

C. Benefit with Redundancy

Our simulations show that redundancy indeed offers considerable latency reduction at a price of additional cost. Fig. 5 shows the distribution of latency of a 10-hop quantum path with different numbers of additional primary swappings (redundancy). The curves are CDFs of E2E latency and the vertical lines in the same color are the corresponding expected latency. When more primary swappings are used (i.e., larger value of r), the CDF is 'higher', which means a higher probability of lower latency. Accordingly, the corresponding vertical lines are placed on left, i.e., the expected latency is lower. Specifically, with $r = 15$ additional primary swappings, the expected latency is reduced from 43.32 time slots ($r = 0$, and gate time $t = 1$ as one unit time slot) to 28.59 time slots.

D. New OLR-P Problem: OLR-PR

Now, we can re-formulate the OLR-R problem to a new problem (OLR-PR), as shown in Equ. (15)-(19), where we now add more duplicates for each path, and each duplicate may use different r redundant primary swapping but up-bounded by R. The decision variable $x_{u,p,r}$ means the number of E2E entanglements established between user pair u over path p with r additional primary swappings, and $t_{u,p,r}$ is the corresponding latency and $\alpha_{u,p,r,e}$ is the number of entanglements used on

edge e. Other constants/inputs and constraints are similar to those in OLR-P but now considered redundant costs.

$$\textbf{OLR-PR}: \min_{x} \sum_{u\in U, p\in P_u, r\in R} t_{u,p,r}\cdot x_{u,p,r} \tag{15}$$

s.t. entanglement requirement

$$\sum_{p\in P_u, r\in R} x_{u,p,r} \geq d_u, \qquad \forall u\in U \tag{16}$$

resource constraints

$$\sum_{u\in U, p\in P_u, e\in p} \alpha_{u,p,r,e} x_{u,p,r} \leq c_e, \quad \forall e\in E \tag{17}$$

$$\sum_{\substack{u\in U, p\in P_u, r\in R \\ e\in p\cap e\in adj(v)}} x_{u,p,r} \leq m_v, \qquad \forall v\in V \tag{18}$$

decision variables

$$x_{u,p,r}\in \mathbb{Z}_0^+. \tag{19}$$

VI. Cost Calculation on Quantum Path

Both OLR-P and OLR-PR problems require the path-swapping solution to give two types of information for their network-wide path selection: (i) expected latency for each path, i.e., $t_{u,p}/t_{u,p,r}$, (ii) expected cost, i.e., edge-level entanglement consumption for each path, $\alpha_{u,p,e}/\alpha_{u,p,r,e}$. We can obtain the latency $t_{u,p}/t_{u,p,r}$ by the formulas for CDFs given in Sections IV and V-A. In this section, we show how to calculate the cost $\alpha_{u,p,e}/\alpha_{u,p,r,e}$.

First, for $\alpha_{u,p,e}$, recall that it means the number of entanglements required on edge e by the path p between user pair u to generate one single E2E entanglement. Note that we mention $C_v = (C_l + C_r)\cdot S_v$ in Section III, but we cannot use this to calculate $\alpha_{u,p,e}$ given the request number d_u. Instead, to obtain one entanglement for a path fraction, we have

$$N_l = N_r = S_v = G(q_v). \tag{20}$$

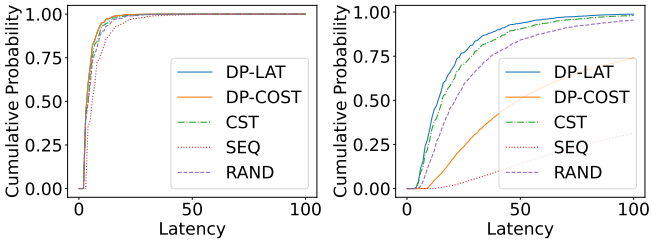
Here, N_l/N_r is the number of entanglements we generated on left/right subpaths. N_l/N_r is obviously the same as the number of attempts we need to succeed once for swapping, which is exactly S_v. Then, $E[N_l] = E[N_r] = \frac{1}{q_v}$. If more entanglements on the path fraction are needed, we can simply multiply $\frac{1}{q_v}$ by the number of copies required. We can repeat this calculation down from E2E entanglement to subpaths until reaching edges to obtain $\alpha_{u,p,e}$. Such calculation is similar to the one in [25].

For $\alpha_{u,p,r,e}$, we can first obtain $\alpha_{u,p,e}$, then add costs for redundant primary entanglements to obtain $\alpha_{u,p,r,e}$. Assume we have obtained $\alpha_{u,p,r,e}$ for $\forall e\in p$, and now we want to add one more redundant primary swapping, conducted on node v whose left and right edges are e_l and e_r. Then, we have

$$\alpha_{u,p,r+1,e_l} = \alpha_{u,p,r,e_l} + \frac{1}{q_v}, \tag{21}$$

$$\alpha_{u,p,r+1,e_r} = \alpha_{u,p,r,e_r} + \frac{1}{q_v}, \tag{22}$$

and $\alpha_{u,p,r+1,e} = \alpha_{u,p,r,e}$ for irrelevant edges $\forall e\notin\{e_l, e_r\}$. We always start from $r=0$ as the base case, where $\alpha_{u,p,0,e} = \alpha_{u,p,e}$. After that, by applying the above recursion on $r>0$, we can obtain $\alpha_{u,p,r,e}$ for $\forall r\in R$.



(a) over 5-hop paths (b) over 15-hop paths

Fig. 6. CDFs of achieved latency by all path-level swapping methods over random paths with different lengths.

VII. Evaluations

We have validated our statistical modeling of latency in Section IV-C and the latency reduction with redundancy strategy in Section V-C. Therefore, here, we focus on the comparison among existing path solutions (swapping schemes), as well as how they perform when integrated with the network scheduler, i.e., the objective of the OLR-P/OLR-PR problem, and whether redundant resources help. Gurobi [32] is used as the ILP solver for the OLR-P and OLR-PR problems.

TABLE I
Network Settings.

Topology	$\lvert V\rvert/\lvert E\rvert$	$\lvert D\rvert$	d_u	c_e	m_v	R
EEnet	12 / 12	6	$[1,1]$	$[10,100]$	$[20,200]$	5
Noel	19 / 25	17	$[1,5]$	$[100,300]$	$[200,600]$	10
Renater	37 / 48	67	$[1,10]$	$[300,500]$	$[600,1000]$	15

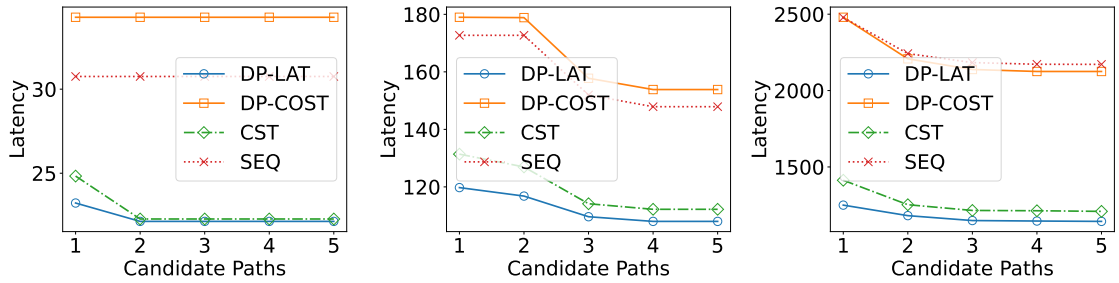
A. Simulation Settings

We consider three real-world network topologies of different sizes from the Internet Topology Zoo [33]: EEnet (small), Noel (medium), and Renater (large), as specified in Table I, for $G(V,E)$. The parameters of quantum resources (c_e, m_v), user demands ($|D|$ and d_u), and maximum number of redundant primary swappings (R) are also given in Table I if not otherwise mentioned. Here $[a,b]$ shows the range of a uniform distribution used for that parameter. By default, we set swapping success probability to $q_v\in[0.5,1]$ uniformly for any node v, and use the swapping gate time as the unit time, i.e., $t=1$. For the number of candidate paths used for each demand (denoted by β) in our path-based optimization problem, we choose 5 as the default value. The number of user pairs is set to $\frac{1}{10}$ of total possible node pairs in the corresponding network. Those networks are also used by [25].

B. Baselines

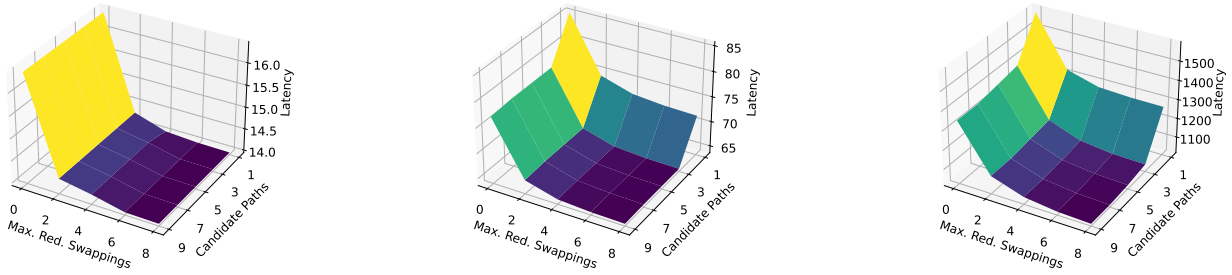
We compare two variations of our proposed DP-based optimal path solution (Section IV) with the other three existing swapping methods.

1) **DP-LAT**: Our DP algorithm with latency as the criteria.
2) **DP-COST**: A variation of our proposed DP algorithm using cost as the criteria instead of latency. DP-COST achieves the optimal cost with less resource for each path, but may not achieve optimal latency for each path.
3) **SEQ**: Sequential swapping used by most existing works (e.g. [9], [14], [15], [18], [19]): the swapping is applied successively along the path from first hop to the last.



(a) EEnet (b) Noel (c) Renater

Fig. 7. Latency of different methods for OLR-P problem in networks with different network scales and different numbers of candidate paths β.



(a) EEnet (b) Noel (c) Renater

Fig. 8. Latency of our method for OLR-PR problem in networks with different scales, max no. of redundant swappings R, and no. of candidate paths β.

4) **CST**: An optimal cost path solution by [25], which performs the swapping based on a complete swapping tree (CST), i.e., a swapping tree with the smallest height. CST is proven to be optimal of cost for the path with homogeneous quantum repeaters (same q_v).
5) **RAND**: Randomly generated swapping sequence.

C. Path Level Performances

We first show the most relevant comparison to our optimization goal: what latency can the baselines achieve on a single quantum path? Fig. 6 shows the CDFs of latency achieved by all methods on random paths with three different lengths. For all cases, our method DP-LAT achieves the best curve, i.e., the 'highest' one (so stochastically dominated by all others). DP-COST can achieve similar latency with DP-LAT when the path has 5-hop. However, when the path has 15-hop, clearly it cannot achieve the best latency. CST is no longer optimal in our settings because it is only optimal for the cost over homogeneous paths (same q_v on all nodes). Another critical observation is that while SEQ is used as the default method in many works [9], [15], [18], [19], it is the worst in all settings. [25] also shows that SEQ has the worst cost over the homogeneous path. Much evidence [15], [34] also suggests that SEQ does not perform well or even may be the worst for many metrics (e.g., latency, cost). Overall, our proposed DP method can achieve the best latency in all cases.

D. Network Wide Performances

We also compare all methods when they are integrated with the network scheduler, i.e., used as the path solutions for OLR-P. Here we test them on three different network topologies (EEnet, Noel and Renater) and choose different numbers of candidate paths β for each user demand. Fig. 7 shows the achieved average latency for each method. Obviously, their performances at the network level are similar to those at the path level. Our DP-LAT also achieves the lowest network-wide latency, while DP-COST and SEQ are generally significantly worse than DP-LAT and CST. With more candidate paths, usually all methods can achieve smaller latency (as in Fig. 7(b) and (c)). But when the network is small (such as EEnet in Fig. 7(a)), the latency does not change much due to its limited paths and resources. Last, within larger network the total latency is larger, which is reasonable since paths in the larger network are usually longer.

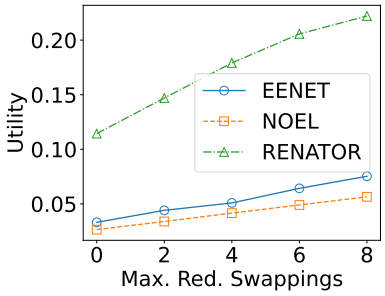
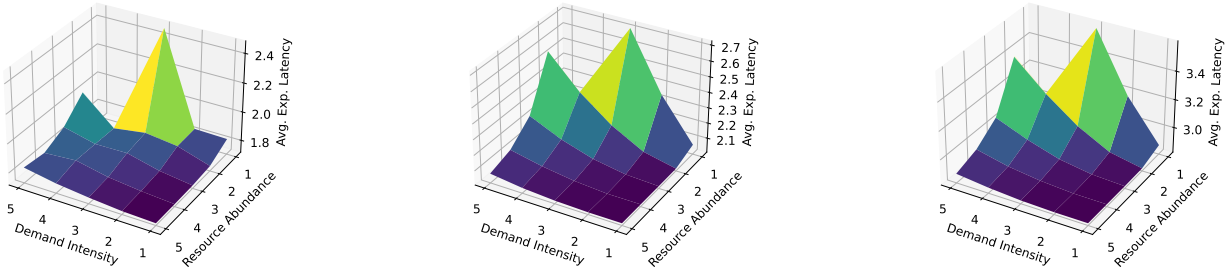


Fig. 9. Resource utility of our method for OLR-PR problem in networks with different scales and max numbers of redundant primary swappings R.

E. Performances with Redundant Resources

Above results confirm that DP-LAT is generally the best for reducing latency at both path and network levels. Now, we explore how our DP-LAT performs for the OLR-PR problem given different initial parameters when redundant resources are available. Because all other methods do not consider redundant



(a) EEnet (b) Noel (c) Renater

Fig. 10. Average latency of our method for OLR-PR problem in networks with different network scales, resource abundances γ, and demand intensity δ.

resources, we mainly show how our redundant design (Section V) can effectively utilize those additional resources. Again we conduct simulations for the three networks at different scales and use resources specified in Table I.

We know that both R (the maximum redundant primary swapping number) and β (the candidate path number between each user pair) can affect the objective of OLR-PR given the same network G, so we show the achieved latency for various R and β in Fig. 8(a)-(c). From Fig. 8(a)-(c), we can have the following observations. First, it is clear that, when we have more candidate paths, we can always have lower latency as we have a larger optimization space. The only exception is for EEnet, the small network, the change of candidate path number does not significantly affect the latency. This is due to the network is too small and there are few simple paths between two nodes. Second, given fixed resources, when we allow more redundant primary swappings, our method can better utilize the idle resources to further reduce latency. This is not possible without our redundant swapping design. Last, this reduction of latency by using redundant resources becomes more beneficial with larger networks. Note that the reduction of latency in EEnet is 14.80% when $R = 8$ (compared to $R = 0$ at the same default $\beta = 5$) while the reduction in Renater is 24.11% for the same R (and β).

From Fig. 8(a)-(c), we see that larger R allows lower latency even for the same resources in the network. Undoubtedly, this is because we are able to utilize more resources (so fewer idle resources) in the network. Therefore, we also investigate the resource utilization for different settings. Here, the resource utilization is the number of used entanglements and memory slots on all edges and nodes divided by the number of all available entanglements and memory slots in the whole network. Fig. 9 reports the results at the default $\beta = 5$. Clearly, in all networks, larger R always leads to higher resource utilization. Note that the relatively low utility is caused by the topology. Those real-world networks have poor connectivity compared to the grid or other generated random topology: they have many line-like or star-like subgraphs, which limit the number of simple paths between node pairs. The problem can be sometimes infeasible if we reduce the available resources to half or double the demands. Renater has better connectivity, so its utilization is relatively higher compared to the other two.

F. Resource Abundance and Demand Intensity

Last but not least, we show how the available resource and demand intensity affect the average expected latency in the three scales of networks. Here, both resource abundance and demand intensity are defined as the multipliers on the default values of corresponding settings. That is, give resource abundance γ and demand intensity δ, the real edge capacity is set to γc_e and node memory is set to γm_v. Similarly, the range of requested entanglement pair for each user is changed to $[\delta, \delta]$, $[\delta, 5\delta]$, and $[\delta, 10\delta]$ for EEnet, Noel, and Renater, respectively. Fig. 10 shows the average expected latency, defined as the OLR-PR objective divided by the total number of user requests.

From Fig. 10, generally, we can see that (i) for fixed demand intensity, more available resources in the network allow lower average latency because we can select more paths with more resource usage but lower latency; (ii) for fixed available resource, higher user demand intensity leads to higher latency as we have to select paths with lower cost and higher latency. (iii) larger network also leads to higher latency. This is because, generally, one can expect that the latency is related to $\log_2 |\pi|$ (the height of balanced swapping trees). Larger graph implies larger path length $|\pi|$, therefore larger latency. Some points are missing because the settings are infeasible, i.e., the available resource is insufficient for the high demands.

VIII. Conclusion

In this paper, we study the critical challenge of latency minimization in QNs at both path and network levels. By modeling the stochastic behavior of entanglement swapping, we developed a DP algorithm for the path-level Optimal Latency Swapping problem, leveraging the proven optimal substructure of E2E latency. Then we introduced a path-selection based optimization and integrated our DP solution with ILP solver to tackle the network-wide Optimal Latency Routing problem. We also proposed a greedy algorithm to allocate redundant resources to further improve latency. Extensive simulations confirm that our methods outperform existing approaches and validate the benefits of redundant entanglement provisioning. This study provides a new framework for latency optimization in quantum networks, paving the way for more efficient and reliable quantum communication systems.

REFERENCES

[1] Z. Li, K. Xue, J. Li, L. Chen, R. Li, Z. Wang, N. Yu, D. S. Wei, Q. Sun, and J. Lu, "Entanglement-assisted quantum networks: Mechanics, enabling technologies, challenges, and research directions," *IEEE Communications Surveys & Tutorials*, 2023.

[2] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpedek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho *et al.*, "A link layer protocol for quantum networks," in *Proc. of ACM SIGCOMM*, 2019.

[3] Y. Wang, A. N. Craddock, R. Sekelsky, M. Flament, and M. Namazi, "Field-deployable quantum memory for quantum networking," *Physical Review Applied*, vol. 18, no. 4, p. 044058, 2022.

[4] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi, "Quantum Internet: networking challenges in distributed quantum computing," *IEEE Network*, 34(1):137-143, 2020.

[5] J. Calsamiglia and N. Lütkenhaus, "Maximum efficiency of a linear-optical bell-state analyzer," *Applied Physics B*, vol. 72, pp. 67–71, 2001.

[6] M. J. Bayerbach, S. E. D'Aurelio, P. van Loock, and S. Barz, "Bell-state measurement exceeding 50% success probability with linear optics," *Science Advances*, 9(32):eadf4080, 2023.

[7] F. Ewert and P. van Loock, "3/4-efficient bell measurement with passive linear optics and unentangled ancillae," *Physical review letters*, vol. 113, no. 14, p. 140403, 2014.

[8] A. Kamimaki, K. Wakamatsu, K. Mikata, Y. Sekiguchi, and H. Kosaka, "Deterministic bell state measurement with a single quantum memory," *npj Quantum Information*, vol. 9, no. 1, p. 101, 2023.

[9] S. Pouryousef, H. Shapourian, A. Shabani, and D. Towsley, "Quantum network planning for utility maximization," in *Proc. of the 1st Workshop on Quantum Networks and Distributed Quantum Computing*, 2023.

[10] A. Chang and G. Xue, "Order matters: On the impact of swapping order on an entanglement path in a quantum network," in *Proc. of IEEE INFOCOM Workshops*, 2022.

[11] T. Coopmans, S. Brand, and D. Elkouss, "Improved analytical bounds on delivery times of long-distance entanglement," *Physical Review A*, vol. 105, no. 1, p. 012608, 2022.

[12] E. Shchukin, F. Schmidt, and P. van Loock, "Waiting time in quantum repeaters with probabilistic entanglement swapping," *Physical Review A*, vol. 100, no. 3, p. 032322, 2019.

[13] M. Ghaderibaneh, C. Zhan, H. Gupta, and C. Ramakrishnan, "Efficient quantum network communication using optimized entanglement swapping trees," *IEEE Trans. on Quantum Eng.*, 3:1–20, 2022.

[14] M. G. de Andrade, C. Zhan, H. Gupta, and C. Ramakrishnan, "On the analysis of quantum repeater chains with sequential swaps," *arXiv preprint arXiv:2405.18252*, 2024.

[15] S. Pouryousef, N. K. Panigrahy, and D. Towsley, "A quantum overlay network for efficient entanglement distribution," *arXiv preprint arXiv:2212.01694*, 2022.

[16] H. Gu, R. Yu, Z. Li, X. Wang, and F. Zhou, "ESDI: Entanglement scheduling and distribution in the quantum internet," *arXiv preprint arXiv:2303.17540*, 2023.

[17] A. Farahbakhsh and C. Feng. "Opportunistic routing in quantum networks," *INFOCOM*, 2022.

[18] Y. Zhao, G. Zhao, and C. Qiao, "E2E fidelity aware routing and purification for throughput maximization in quantum networks," in *IEEE INFOCOM*, 2022.

[19] J. Li, M. Wang, K. Xue, R. Li, N. Yu, Q. Sun, and J. Lu, "Fidelity-guaranteed entanglement routing in quantum networks," *IEEE Trans. on Comm.*, 70(10):6748–6763, 2022.

[20] Z. Jia and L. Chen, "From entanglement purification scheduling to fidelity-constrained multi-flow routing," *arXiv preprint arXiv:2408.08243*, 2024.

[21] W. Dai, T. Peng, and M. Z. Win, "Optimal remote entanglement distribution," *IEEE JSAC*, vol. 38, no. 3, pp. 540–556, 2020.

[22] H. Gu, Z. Li, R. Yu, X. Wang, F. Zhou, and J. Liu, "FENDI: High-fidelity entanglement distribution in the quantum internet," *arXiv preprint arXiv:2301.08269*, 2023.

[23] A. S. Cacciapuoti, M. Caleffi, R. Van Meter, and L. Hanzo, "When entanglement meets classical communications: quantum teleportation for the quantum Internet," *IEEE Transactions on Communications*, 68(6):3808-3833, 2020.

[24] M. Viscardi, J. Illiano, A. S. Cacciapuoti and M. Caleffi, "Entanglement distribution in the quantum Internet: an optimal decision problem formulation," in *Proceedings of 2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2023.

[25] J. Liu, X. Liu, X. Wei, and Y. Wang, "Topology design with resource allocation and entanglement distribution for quantum networks," in *IEEE SECON*, 2024.

[26] X. Wei, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Optimizing satellite-based entanglement distribution in quantum networks via quantum-assisted approaches," in *Proceedings of IEEE International Conference on Quantum Communications, Networking, and Computing (QCNC 2024)*, 2024.

[27] X. Wei, J. Liu, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Optimal entanglement distribution problem in satellite-based quantum networks," *IEEE Network*, vol. 9, no. 1, p. 97-103, 2025.

[28] X. Wei, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Entanglement from sky: Optimizing satellite-based entanglement distribution for quantum networks," *IEEE/ACM Transactions on Networking*, vol. 32, no. 6, p. 5295-5309, 2024.

[29] J. Liu, L. Fan, Y. Guo, Z. Han, and Y. Wang, "Co-design of network topology and qubit allocation for distributed quantum computing," in *Proceedings of IEEE International Conference on Quantum Communications, Networking, and Computing (QCNC 2025)*, 2025.

[30] Y. Lee, E. Bersin, A. Dahlberg, S. Wehner, and D. Englund, "A quantum router architecture for high-fidelity entanglement flows in quantum networks," *npj Quantum Information*, vol. 8, no. 1, p. 75, 2022.

[31] Y. Zhao and C. Qiao, "Redundant entanglement provisioning and selection for throughput maximization in quantum networks," in *Proc. of INFOCOM*, 2021.

[32] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: https://www.gurobi.com

[33] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE JSAC*, 29(9):1765 –1775, 2011.

[34] N. K. Panigrahy, T. Vasantam, D. Towsley, and L. Tassiulas, "On the capacity region of a quantum switch with entanglement purification," *arXiv preprint arXiv:2212.01463*, 2022.